

# ANNEX

(2876-2: OnTheBus)

## Índex de continguts

1	Annex A: Anàlisi de botons dels terminals .....	3
2	Annex B: Proves i Resultats Simulats .....	8
3	Annex C: Símbols Gestures .....	10
3.1	Símbols d'edició.....	10
3.2	Mode text.....	11
3.3	Mode numèric .....	13
4	Annex D .....	14
4.1	Views .....	14
4.1.1	TextView .....	14
4.1.2	EditText.....	14
4.1.3	Button.....	15
4.1.4	ImageButton .....	15
4.1.5	CheckBox .....	15
4.1.6	ToggleButton .....	16
4.1.7	RadioGroup i RadioButton .....	17
4.1.8	Spinner .....	17
4.1.9	AutoComplete .....	18
4.1.10	MapView .....	19
4.2	VIEWGROUPS .....	19
4.2.1	LinearLayout .....	19
4.2.2	RelativeLayout .....	20
4.2.3	TableLayout .....	20
4.2.4	ListView .....	21

## 1 Annex A: Anàlisi de botons dels terminals

Taula sobre l'anàlisi de la disposició dels botons en els diferents terminals mòbils d'*Android*.

<u>Model del terminal</u>	<u>Botons físics</u>	<u>Botons tàctils</u>
HTC Dream	Trucada Home Menú Back Penjar trucada/encendre i apagar el terminal Volum Trackball Càmera Teclat Qwerty	
HTC Màgic	Trucada Home Menú Back Penjar trucada/encendre i apagar el terminal Volum Trackball	
HTC Hero	Trucada Home Menú Back Buscar Penjar trucada/encendre i apagar el terminal Volum Trackball	
HTC Nexus One	Volum Trackball Encendre i apagar el terminal	Back Menu Home

		Buscar
Samsung Nexus S	Volum Encendre i apagar el terminal	Back Menu Buscar Home
HTC Wildfire	Volum Encendre i apagar el terminal	Home Menú Back Buscar Trackball Òptic
HTC Desire	Volum Encendre i apagar el terminal	Home Menú Back Buscar Trackball Òptic
HTC Wildfire S	Volum Encendre i apagar el terminal	Home Menú Back Buscar
HTC Desire S	Volum Encendre i apagar el terminal	Home Menú Back Buscar
HTC Desire Z	Volum Encendre i apagar el terminal Càmera Teclat Qwerty	Home Menú Back Buscar Trackball Tàctil
Samsung Galaxy S	Volum Encendre i apagar el terminal	Menú Home

		Back
Samsung Galaxy S II	Volum Encendre i apagar el terminal	Menú Home Back
Samsung Galaxy mini	Volum Encendre i apagar el terminal	Menú Home Back
Samsung Galaxy Ace	Volum Encendre i apagar el terminal	Menú Home Back
Samsung Galaxy Pro	Volum Encendre i apagar el terminal Teclat Qwerty	Menú Home Back Buscar
Huawei U8110 (Ivy)	Volum Encendre i apagar el terminal Tecla de navegació "Trackball" Tecla OK Trucada Penjar Trucada Càmera	Home Menú Back
Huawei U8500	Trucada Menú Back Penjar Trucada Volum Encendre i apagar el terminal	Trackball òptic
Huawei U8650 (Sonic)	Volum Encendre i apagar el terminal	Home Menú

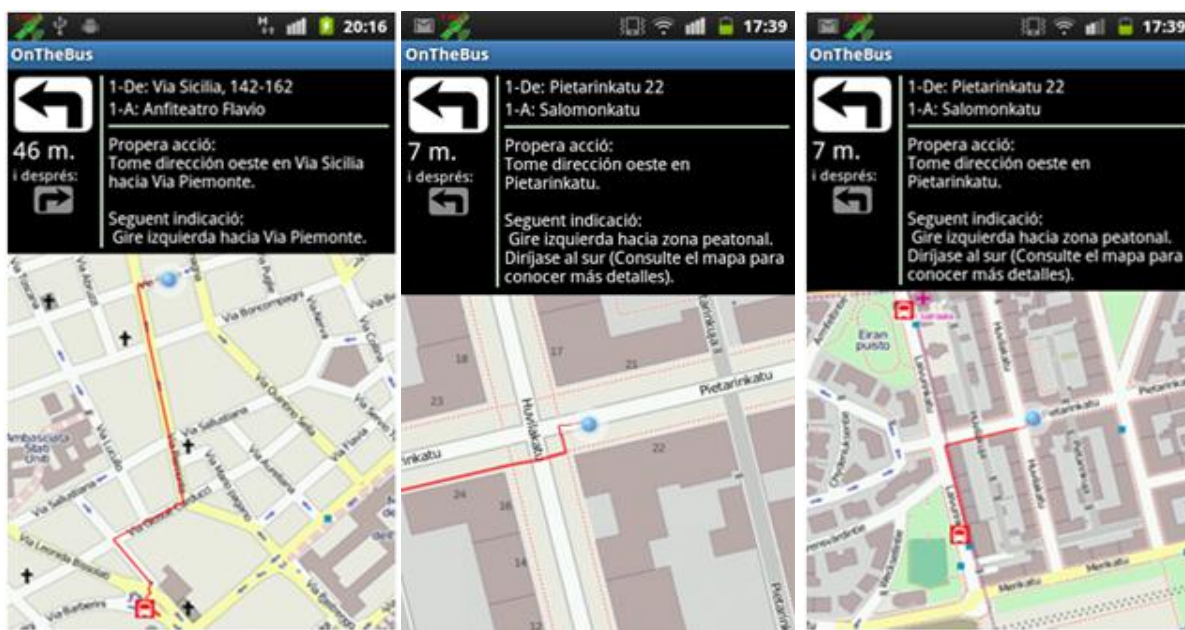
		Back Buscar
<b>LG Optimus Me P350</b>	Tecla Enviar Encendre i apagar el terminal / Bloquejar Volum	Menú Home Back Buscar
<b>LG Optimus One P500</b>	Encendre i apagar el terminal / Bloquejar Volum Menú Home Back Buscar	
<b>LG Optimus 2X P990</b>	Encendre i apagar el terminal / Bloquejar Volum	Menú Home Back Buscar
<b>ZTE Blade</b>	Encendre i apagar el terminal / Bloquejar Volum Home Menú Back	
<b>Sony Ericsson Xperia X8</b>	Encendre i apagar el terminal / Bloquejar Volum Menú Home Back Càmera	
<b>Sony Ericsson Xperia X10</b>	Encendre i apagar el terminal / Bloquejar Volum Menú Home	

	Back Càmera	
Sony Ericsson Xperia X10 mini Pro	Encendre i apagar el terminal / Bloquejar Volum Menú Home Back Càmera Teclat Qwerty	
Sony Ericsson Xperia Arc	Encendre i apagar el terminal / Bloquejar Volum /Zoom Menú Home Back Càmera	
Sony Ericsson Xperia Ray	Encendre i apagar el terminal / Bloquejar Volum	Back Home Menu

## 2 Annex B: Proves i Resultats Simulats

En aquest annex s'inclouen les diferents proves que s'han fet a les ciutats a les que no s'han pogut fer proves de camp. Per poder testejar els diferents mòduls s'utilitza *Fake GPS*, una aplicació disponible al *Android Market* i que permet establir una falsa localització GPS en les diferents aplicacions disponibles al mòbil, per tant, mitjançant aquest programa es pot simular que estem a qualsevol ciutat del món.

- **Rutes Walk**

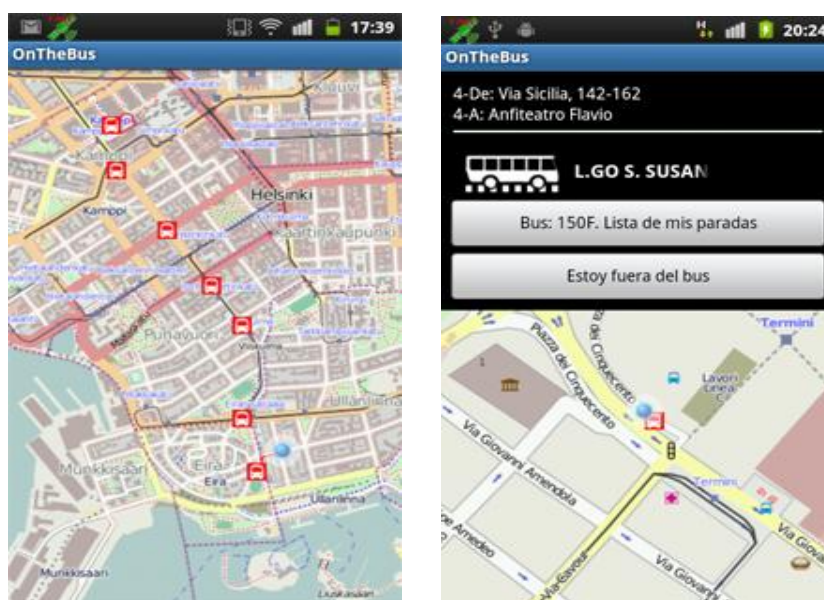


Il·lustració 1: Exemple a Roma;

Il·lustració 2: Ruta a Helsinki;

Il·lustració 3: Trajecte a Helsinki;

- **Rutes Bus**



Il·lustració 4: Bus a Helsinki;

Il·lustració 5: Exemple a Madrid;



- Rutes amb transbord

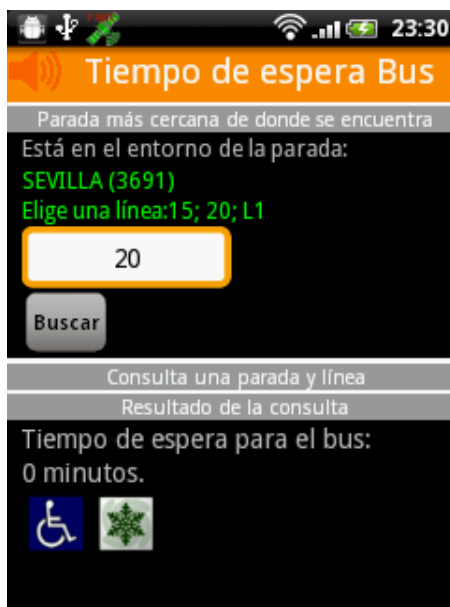


Il·lustració 6: Transbord a Madrid;

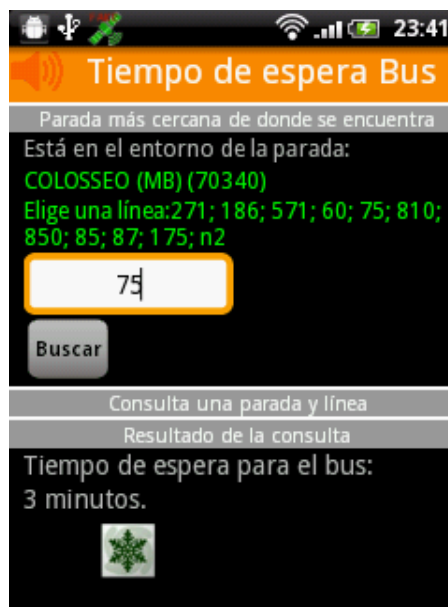


Il·lustració 7: Transbord a Roma;

- Temps d'Espera



Il·lustració 8: Temps Espera a Madrid;



Il·lustració 9: Temps Espera a Roma;

### 3 Annex C: Símbols Gestures

*Després de moltes variacions, símbols utilitzats a l'alfabet finalista.*

#### 3.1 Símbols d'edició

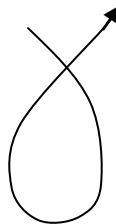
Els símbols s'han de realitzar seguint el sentit de les fletxes. Un sol traç per *Gesture*. Per esborrar tot el contingut, botó "Menú" -> "Esborrar".



**Esborrar últim caràcter**

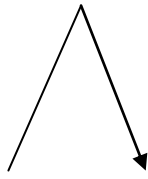


**Espai**



**Canvi Lletres / Números**

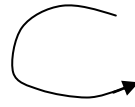
3.2 Mode text



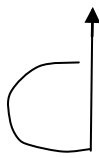
A



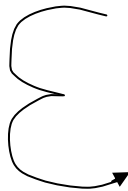
B



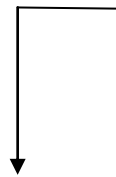
C



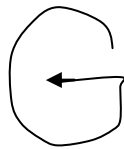
D



E



F



G



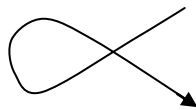
H



I



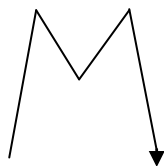
J



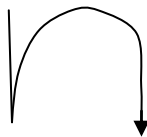
K



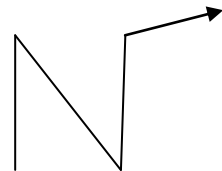
L



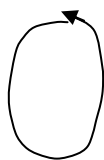
M



N



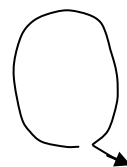
Ñ



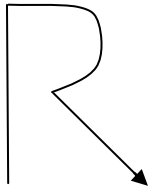
O



P



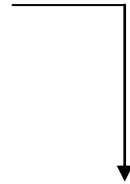
Q



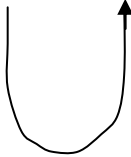
R



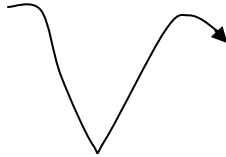
S



T



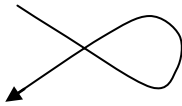
U



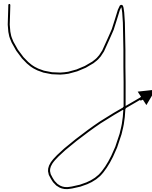
V



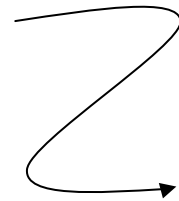
W



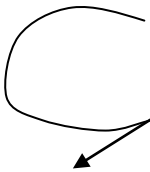
X



Y

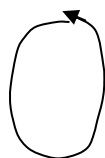


Z



Ç

### 3.3 Mode numèric



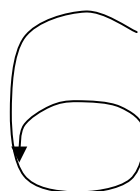
0



5



1



6



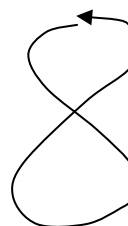
2



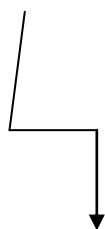
7



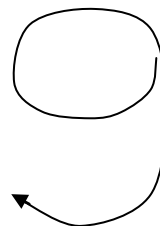
3



8



4



9

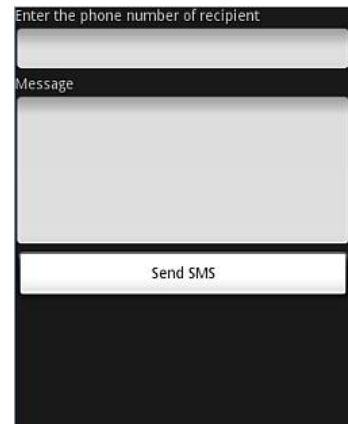
## 4 Annex D

### 4.1 Views

#### 4.1.1 TEXTVIEW

És senzillament, un lloc per posar text. No permet modificar gaire la seva aparença perquè per això ja existeixen altres tipus més especialitzats, com els *EditText*. No obstant, té atributs que podem modificar, com són la seva alçada, amplada, tipus de lletra, color de text, color de fons, etc...

```
<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="el meu text"
    android:id="@+id/my_txt"/>
```



Imatge 1 - Exemple de TextView, EditText i Button

En cas de voler modificar el seu contingut en temps d'execució, hauríem de crear una variable associada al seu identificador a l'xml, d'aquesta manera:

A l'onCreate per exemple:

```
txt = (TextView) findViewById(R.id.my_txt);
```

i després, per modificar el text inicial:

```
txt.setText("Text de prova");
```

#### 4.1.2 EDITTEXT

És una subclasse de TextView, amb la particularitat de que permet editar el seu contingut a l'usuari.

```
<EditText
    android:id="@+id/txtName"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"/>
```

S'ha de vincular a una variable...

```
EditText myEdit = (EditText) findViewById(R.id.txtName);
```

... per tal de poder obtenir el seu text:

```
String myText= myEdit.getText().toString();
```

#### 4.1.3 BUTTON

---

Representa el típic botó que es pot apretar i està lligat a certs estats.

```
<Button
    android:id="@+id/my_but"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Acceptar"/>
```

Per la seva naturalesa, generalment el vinculem a una variable:

```
Button myBut = (Button) findViewById(R.id.my_but);
```

Necessitem assignar-li un *listener* per tal de poder capturar l'event de pulsació.

```
myBut.setOnClickListener(this);
```

I després al mètode *onClick*, capturem l'event i es realitza una acció:

```
public void onClick(View vista) {
    if(vista.getId() == myBut.getId()){
        Toast.makeText(this, "Has apretat el botó!",
            Toast.LENGTH_LONG).show();
    }
}
```

#### 4.1.4 IMAGEBUTTON

---

És un widget de tipus botó, com l'anterior View però en aquest, en comptes de text, es pot posar imatges. Suporta diferents tipus de format, com jpg, png, etc...

```
<ImageButton
    android:id="@+id/btnImg1"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:src="@drawable/icon"/>
```



Imatge 2 - ImageButton i  
CheckBox

#### 4.1.5 CHECKBOX

---

És un tipus especial de botó però que té dos estats, marcat i desmarcat.

```
<CheckBox
```

```

android:id="@+id/chkbut"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:text="my CheckButton"/>

```

Per tractar-los de manera dinàmica, cal fer:

```

CheckBox c1 = (CheckBox) findViewById(R.id.check1);
CheckBox c2 = (CheckBox) findViewById(R.id.check2);

```

I com abans, al mètode `onClick`:

```

public void onClick(View vista) {
    if(c1.isChecked())
        Toast.makeText(this, "c1 marcado", Toast.LENGTH_LONG).show();
    }else if(c2.isChecked()){
        Toast.makeText(this, "c2 marcado", Toast.LENGTH_LONG).show();
    }
}

```

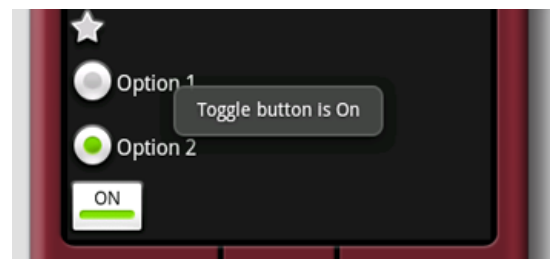
#### 4.1.6 TOGGLEBUTTON

---

```

<ToggleButton
    android:id="@+id/toggle1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textOn = "ON"
    android:textOff = "OFF"/>

```



Imatge 3 - RadioButton, RadioGroup i ToggleButton

Per obtenir el seu valor actual:

```

ToggleButton btnBoton2 = (ToggleButton) findViewById(R.id.BtnBoton2);
btnBoton2. setOnClickListener(this);

```

*i després a `onClick`:*

```

public void onClick(View vista) {
    if(btnBoton2.isChecked())
        Toast.makeText(this, "ENCENDIDO", Toast.LENGTH_LONG).show();
    }else{
        Toast.makeText(this, "APAGADO", Toast.LENGTH_LONG).show();
    }
}

```



#### 4.1.7 RADIOGROUP I RADIOBUTTON

---

Els RadioButton tenen dos estats, marcats i desmarcats. Un cop un RadioButton ha sigut marcat, no es pot desmarcar fins que no es marqui altre RadioButton. El RadioGroup serveix per tenir vinculats un grup de RadioButtons, on només pot estar marcat un.

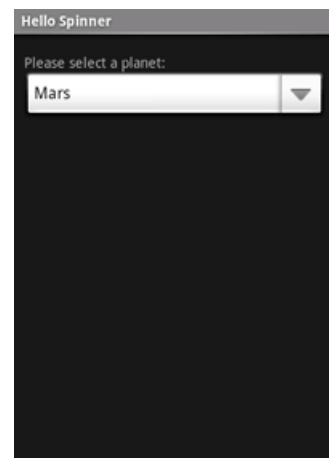
```
<RadioGroup
    android:id="@+id/rdbGp1"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    >
    <RadioButton android:id="@+id/rdb1"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Option 1"/>
    <RadioButton android:id="@+id/rdb2"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Option 2"/>
</RadioGroup>
```

#### 4.1.8 SPINNER

---

És la típica llista desplegable, on l'usuari pot seleccionar una opció. Un cop s'ha seleccionat un element, la llista desapareix i aquest queda fixat.

```
<RelativeLayout
...>
<Spinner
    android:id="@+id/spinner"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:drawSelectorOnTop="true"
    android:prompt="Choose a planet"
/>
</RelativeLayout>
```



Imatge 4 - Spinner

I després cal definir a l'*onCreate* l'ArrayAdapter que estarà vinculat a cada un de l'array d'elements.

```

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);

    Spinner s = (Spinner) findViewById(R.id.spinner);
    ArrayAdapter adapter = ArrayAdapter.createFromResource(
        this, R.array.planets, android.R.layout.simple_spinner_item);

    adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
    s.setAdapter(adapter);
}

```

#### 4.1.9 AUTOCOMPLETE

Es similar a un EditText convencional però aporta la funcionalitat de mostrar a l'usuari consells per autocompletar els texts, mitjançant un desplegable de possibles opcions.

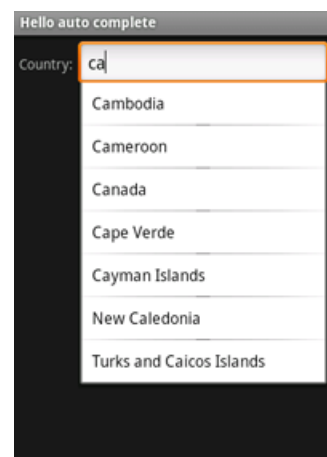
Els consells es reben en una col·lecció de strings que estan vinculats al widget mitjançant un ArrayAdapter que els hi dona el format.

```

<include layout="@layout/qualsevolTipusDeLayout"
...>
<AutoCompleteTextView
    android:id="@+id/autocomplete"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"

    android:layout_marginLeft="5dp"/>
</include>

```



Imatge 5 - AutoComplete TextView

Després a l'onCreate només cal afegir:

```

AutoCompleteTextView textView = (AutoCompleteTextView)
findViewById(R.id.autocomplete)

ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,
R.layout.list_item, COUNTRIES);
textView.setAdapter(adapter);

```

I l'array amb els valors desitjats:

```

static final String[] COUNTRIES = new String[] {
    "Afghanistan", "Albania", "Algeria", "American Samoa", "Andorra", ... }

```

#### 4.1.10 MAPVIEW

quest tipus de *View* serveix, com el seu nom indica, per crear una *Activity* per visionar un mapa. Per defecte, es podria haver utilitzat *Google Maps*, però per temes de llicència, hem preferit fer servir la plataforma *Osmdroid* (*Open Street Maps per Android*), que també ens donava el servei que necessitàvem sense restriccions.

Necessita de permisos especials, com accés a internet o al gps, depenent de l'objectiu, que s'han de declarar al *Manifest.xml*.



Imatge 6 -MapView

La seva implementació requereix una explicació més detallada, per aquesta raó es deixa per més endavant.

### 4.2 VIEWGROUPS

És una agrupació de *Views*, i en si mateixa, també es considera una altra *View*.

#### 4.2.1 LINEARLAYOUT

Aquest *layout* apila un darrera l'altre els elements fills que el formen, i ho fa de manera horitzontal o vertical, segons es defineixi l'atribut *orientation*. En aquest layout es disposa d'un atribut *weight*, que ens permet donar posicions proporcionals vers els altres elements.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"

    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
...views
</LinearLayout>
```

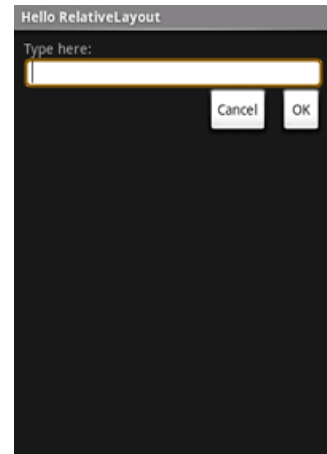


Imatge 7 - LinearLayout

### 4.2.2 RELATIVELAYOUT

La particularitat d'aquest *layout* és que permet especificar posicions relatives a altres Views de la pantalla. Per tant, ens és possible definir per exemple, que el botó *Cancel* de la imatge es trobi sota el *EditText*, a la esquerra del botó *OK*, *alineat a la dreta del layout pare*.

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content">
    <EditText
        android:id="@+id/TxtNombre"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content" />
    <Button
        android:id="@+id/BtnAceptar"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/TxtNombre"
        android:layout_alignParentRight="true" />
</RelativeLayout>
```

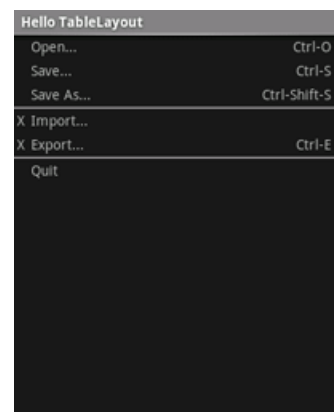


Imatge 8 - RelativeLayout

### 4.2.3 TABLELAYOUT

Aquest layout permet distribuir els elements fills en forma de taula, definint files i columnes, de manera semblant a la definició en HTML. Generalment, es definiran les files (*TableRow*) de la taula, i a l'interior de cada una d'elles, es definirà el número de columnes que hi contindrà insertant-li Views, perquè no hi ha cap element de tipus *TableColumn*, sino que les columnes queden definides pel nombre màxim de Views que s'han insertat en una fila.

```
<TableLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:stretchColumns="1" >
    <TableRow>
        <TextView android:text="casella 1.1"/>
```



Imatge 9 - TableLayout

```

        <TextView android:text="casella 1.2"/>
    </TableRow>
    <TableRow>
        <TextView android:text="casella 2.1"/>
        <TextView android:text="casella 2.2"/>
    </TableRow>
</TableLayout>

```

#### 4.2.4 LISTVIEW

És un tipus de View que mostra els elements seleccionables de forma vertical en una pantalla amb *scroll* (barra de desplaçament) si cal. Els elements s'aniran obtenint d'un *ListAdapter*.

Per començar, la pantalla ha d'extendre de *ListActivity* en comptes d'*Activity*, i al mètode *onCreate*, es fa la vinculació entre l'Array d'*Adapters* de *String*, amb un format prefixat i l'array d'elements.

```

@Override
public void onCreate(Bundle savedInstanceState)
{
    super.onCreate(savedInstanceState);
    setListAdapter(new ArrayAdapter<String>(this,
        android.R.layout.simple_list_item_1, COUNTRIES));
    getListView().setTextFilterEnabled(true);
}

```



Imatge 10 - ListView

En aquest cas no es carrega cap layout xml (en casos més especialitzats, es pot carregar un format per cada fila).

L'array d'elements vindria definit així:

```

static final String[] COUNTRIES = new String[] {
    "Afghanistan", "Albania", "Algeria", "American Samoa", "Andorra"... }

```

A més de tots els elements que hem comentat, també són necessaris altre tipus d'elements com són els menús, les notificacions, etc... però per qüestió d'espai, no ho ampliarem en aquesta ocasió.

A handwritten signature in blue ink, consisting of several overlapping loops and strokes, positioned above a horizontal line.

---

**Signatura: Eric Xavier Lara Amat y León**

**Bellaterra, Setembre 2011**

